

# MEMORY CONTROL METHOD AND APPARATUS

**Patent number:** JP1306917  
**Publication date:** 1989-12-11  
**Inventor:** MUUSUDASHI SHIYAWANKAA MENON  
**Applicant:** IBM  
**Classification:**  
 - international: G06F3/06; G11B20/12  
 - european: G06F3/06F; G06F3/06M; G11B20/12M  
**Application number:** JP19890078721 19890331  
**Priority number(s):** US19880197057 19880520

Also published as:



EP0347032 (A)  
 EP0347032 (A)  
 EP0347032 (B)

Report a data error he

## Abstract of JP1306917

**PURPOSE:** To improve the access performance of a record to be emulated by reforming the data record of a first format to a second non-interchangeable recording format while maintaining the byte displacement relative position of the record of the first format in the second format. **CONSTITUTION:** At the time of correcting so as to be stored by the second format, data of the first format including a control field and a data field is corrected to be stored in each address possible record as data of the second non-interchangeable mode while maintaining the byte displacement address possibility of the record of the first format. Correction like this attains the emulation of the first format to the second format by maintaining a gap between records by both formats while removing a gap within the record, error detecting redundant information, physical parameter data and a padding characteristic to the first format during correcting. Thereby, the access performance of the record to be emulated is improved.

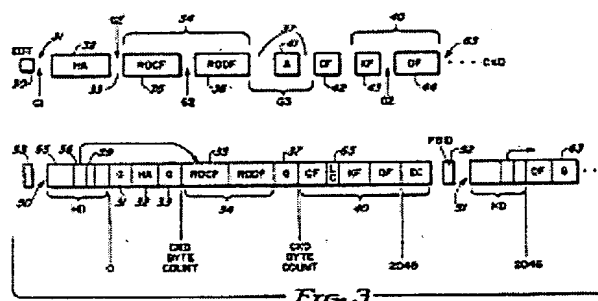


Fig. 3

Data supplied from the esp@cenet database - Worldwide

## ⑫ 公開特許公報(A)

平1-306917

⑤Int.Cl.

G 06 F 3/06  
G 11 B 20/12

識別記号

3 0 1

庁内整理番号

Z-6711-5B  
8524-5D

⑬公開 平成1年(1989)12月11日

審査請求 有 請求項の数 3 (全16頁)

⑭発明の名称 記憶制御方法及び装置

⑯特 願 平1-78721

⑰出 願 平1(1989)3月31日

優先権主張 ⑱1988年5月20日⑲米国(US)⑳197057

⑳発 明 者

ムースダシ・シヤワン  
カー・メノンアメリカ合衆国カリフォルニア州サン・ホセ、モントロ・  
プレス6017番地

㉑出 願 人

インターナショナル・  
ビジネス・マシーン  
ズ・コーポレーションアメリカ合衆国10504、ニューヨーク州 アーモンク(番  
地なし)

㉒代 理 人

弁理士 山本 仁朗 外1名

## 明 細 書

1. 発明の名称 記憶制御方法及び装置

2. 特許請求の範囲

(1) 複数のレコードからなる第1の様式のデータを、該第1の様式のまま、第2の様式の記憶手段に記憶するための、下記ステップ(イ)ないし(ニ)を含む記憶制御方法、ただし、上記第1の様式のデータは、それぞれアドレス可能な第1の制御フィールド及び第1のデータ・フィールドとフィールド間ギャップとを各レコード内に含むと共に、相次ぐレコードの間にレコード間ギャップを含み、各レコードのフィールド及びギャップのアドレス可能な位置は第1の様式のデータの始点を基準とするバイト・カウントで識別可能であり、かつ各レコードの制御フィールドは特定の制御に関する部分を含んでおり、さらに、上記第2の様式の記憶手段は、それぞれ所定のバイト数の記憶容量を有する複数のアドレス可能なブロックを含むものとする：

(イ) それぞれ上記アドレス可能なブロックに等しい記憶容量を有する複数のアドレス可能な領域をバッファ手段に設定し、

(ロ) 上記第1の様式のデータから、上記特定の制御に関する部分及びフィールド間ギャップを除去し、

(ハ) 上記第1の様式の残りのデータ及びレコード間ギャップを上記バッファ手段の複数のアドレス可能な領域に記憶し、

(ニ) 上記記憶手段のアドレス可能な各ブロックが上記バッファ手段の対応する領域からのデータを受け取り、かつ上記第2の様式での始点から各レコードの先頭までのバイト変位が上記第1の様式のデータの始点を基準とするカウント・バイトに等しくなるように、上記バッファ手段内のデータ及びレコード間ギャップを上記記憶手段に記憶する。

(2) 制御フィールド、データ・フィールド及びフィールド間ギャップをそれぞれ有する複数のレコードとレコード間ギャップとを含みかつ上記

フィールド内に誤り訂正、充填、及び物理的パラメータのための特定部分を含むデータを供給するデータ供給手段と、所定数のバイトをそれぞれ記憶しうる複数のアドレス可能な固定ブロックを有する記憶手段とを有する情報処理システムにおいて、上記記憶手段におけるデータの記憶を制御するための、下記ステップ(a)ないし(e)を含む記憶制御方法：

(a) 上記データ供給手段によって供給されるデータから上記特定部分及びフィールド間ギャップを除去し、

(b) 上記データの最後のレコードの制御フィールドに最終レコード標識を付加し、

(c) 上記データにおける順序と同じ順序でかつ上記レコード間ギャップを維持しつつ上記データの複数のレコードを配列し、その際、配列後のデータの始点から各レコードの先頭までのバイト単位の変位が元のデータにおける始点から各レコードまでのバイト単位の変位に等しくなるようにし、

(d) 配列したデータを、元のデータの形式に

- 3 -

レータ手段が、

上記所定の記憶容量に等しい記憶容量をそれぞれ有する複数の領域を有するバッファ手段に上記カウント・キー・データ形式のデータを一時的に記憶し、そこから上記記憶装置へ転送する手段と、

仮想トラックの始点からカウント・キー・データ形式の各レコードの先頭までのバイト変位を、カウント・キー・データ形式のトラックの始点から各レコードの先頭までのバイト変位と等しくするようにして、上記データを上記記憶装置の複数の固定ブロックに記憶するための仮想トラックを設定する手段と、

上記仮想トラックの外において、各固定ブロックの先頭にヘッダを設け、該固定ブロックにカウント・キー・データ形式の第1のレコードが含まれるときには、該第1のレコードまでの変位を示すポインタを上記ヘッダに記録し、該固定ブロックにカウント・キー・データ形式の第1のレコードが含まれないときには、そのことを示す情報を上記ヘッダに記録する手段と、

- 5 -

かかわりなく、上記記憶手段の固定ブロックに対応する長さに分割し、

(e) 分割したデータを上記記憶手段に書き込む。

(3) それぞれ所定の記憶容量を有する複数の固定ブロックを有する固定ブロック形式の直接アクセス型記憶装置と、カウント・キー・データ形式に関連したプログラムを用いる上位プロセッサと、該上位プロセッサと上記記憶装置との間に設けられたチャンネル・プロセッサとを有しかつ該チャンネル・プロセッサが上記上位プロセッサと上記記憶装置との間のデータ転送及びそれに関連した動作を命じる複数のチャンネル・コマンドを生じる制御部を有する情報処理システムにおいて、上記記憶装置を制御する装置であって、

上記チャンネル・プロセッサの制御部と上記記憶装置との間に設けられていて、所定のチャンネル・コマンドに回答して、上記記憶装置においてカウント・キー・データ形式のデータの記憶処理をエミュレートするエミュレータ手段を有し、該エミュ

- 4 -

を有する記憶制御装置。

### 3. 発明の詳細な説明

#### A. 産業上の利用分野

本発明は、情報処理システムの記録動作に関し、具体的には、第1の様式とは互換性のない第2の様式でフォーマットされた記録装置に第1の様式で配列されたデータを記録することに関する。

#### B. 従来技術及び発明が解決しようとする問題点

情報処理システムは、周辺データ記録装置にデータを記録するために多様なモードを使用してきた。あるデータ記録モードが選択されると、情報処理システムのプログラミングが、選択されたモードで動作するように設計される。この制限は、データ記録装置が、プログラミングで企図されたモードを使用しなければならず、またその逆もあり得ることを意味している。ある記録モードを他の記録モードでエミュレートすることが実現されているが、こうしたエミュレーションでは、プログラミングがエミュレートされたモードを容易にアドレスできなかった。すなわち、エミュレーション・ソフトウェア

- 6 -

は、第1の様式のアドレスを第2の様式の異なるアドレスに変換しなければならなかった。こうしたアドレス変換は、第2の様式に記録されたとき第1の様式のレコード・アドレスを混乱させるだけでなく、かなりのバッファリング資源と命令実行資源を必要とする。

マリリン・ボール (Marilyn Bohl) 著「IBM 直接アクセス記憶装置入門 (INTRODUCTION TO IBM DIRECT ACCESS STORAGE DEVICES)」、出版物SR20-4738、Science Research Associates, Inc.、シカゴ、の第4章には、2つのデータ様式が記載されている。第1の様式は、インターナショナル・ビジネス・マシーンス社 (IBM) のディスク記録装置 (ディスク・ファイル及びDASDとも呼ばれる) で使用される周知のカウント、キー、データ (CKD) 様式である。多くの大規模情報処理システムが、そのプログラミング及び記録にCKDデータ様式を利用している。ボールはまた、固定ブロック (FB) 記録 (ブロック・アーキテクチャ (FBA) 様式と

- 7 -

以前のFBA装置でCKD様式データを記録する手順は、「IBM4321/4331プロセッサ互換性機能 (IBM 4321/4331 Processors Compatibility Features)」、IBM出版物GA33-1528-2、第3版、1982年9月、に一般的に記載されている。この様式重畳記録では (真のエミュレーションではない)、IBM2311及び2314DASDボリューム上のCKD様式データが、IBM3310FBA記録装置に記録された。この出版物の教示によると、CKD様式データをFBA記録装置に圧縮するため、すべてのギャップ (DASDに記録されたときデータ様式の様々なフィールド間に挿入されたデータを含まない領域) が除去される。この動作は、CKD様式によるレコードのCKDアドレス可能性を混乱させる。CKD記録装置のトラック全体 (そのトラックがデータで満たされていない場合でも) がFBA記録装置に記録される。様式重畳記録のマッピングが、前記の出版物の第2-3ページに示されている。FBA記録装置に記憶された

- 9 -

も呼ばれる) も記載している。対照的に、CKD様式では、レコード・サイズに応じてディスク・トラック記録様式を分割することにより、大きなレコードのディスク空間を効率よく使用できる。FBA様式では同一サイズの1組のアドレス可能領域 (ブロックと呼ばれる) を使用し、その領域のそれぞれに、512バイト、1024バイト、2048バイトなどある固定量までのデータが記憶できる。(512バイトより小さいものなど) 小さいサイズのレコードでは、FBAはCKDに比べていくつかの利点をもつ。いずれにせよ、一部はCKD様式を使用し一部はFBA様式を使用するDASDを備えた情報処理システムが利用される。システムで実行されるプログラムの多くは、CKD様式で動作するように設計されている。したがって、CKD様式データをFBA装置で記録できるようにすることが望ましい。FBA様式で記憶されたすべてのデータ・レコードのCKDレコード・アドレス可能性を維持することも望ましい。

- 8 -

CKDレコードのCKD回転アドレス可能性を維持することにより、FBA様式の記録装置でCKD様式の真のエミュレーションを実現することが望ましい。

本発明の目的は、第2様式において第1様式のレコードのバイト変位相対位置を維持しながら、第1の様式のデータ・レコードを第2の非互換レコード様式にフォーマットし直すことにある。

本発明の他の目的は、第2の様式でのアドレッシングで仮想トラック概念を使用して、両方の様式でレコードのバイト変位アドレスを同じに維持しながら、制御情報を第2の様式に追加することにより第2の非互換様式で記録されたときに第1の様式のデータのアドレッシングを提供することである。

#### C. 問題点を解決するための手段

本発明によれば、第2の様式で記憶されるように修正されたときに第1の様式のレコードのバイト変位アドレス可能性を維持しながら、各アドレス可能レコード中に (CKD様式データ中のカウ

- 10 -

ント・フィールドやキー・フィールドなどの) 制御フィールドとデータ・フィールドを含む第1の様式のデータが、第2の非互換様式のデータとして記憶されるように修正される。こうした修正は、その間に、レコード内ギャップ、エラー検出冗長情報、物理パラメータ・データ、第1の様式に固有のパディングを除去しながら、両方の様式でレコード間ギャップを維持することにより、第1の様式の第2の様式へのエミュレーションを達成する。第2の様式でのデータ構成は、第1の様式から独立し、レコードが第2の様式のアドレス可能部分に分割されるようになっている。

制御情報は、アドレス可能部分に記憶された1つのレコードの1つの制御フィールドに関係する第2の様式の各アドレス可能部分に追加される。第1の様式のデータ長は完全に可変であり、最小数のアドレス可能部分が修正された第1の様式のデータを第2の様式で記憶する際に消費される。

本発明の特定の実施例では、CKD様式データの1つまたは複数のDASDトラックが、FBA

- 11 -

入出力マシン動作を実行するため、主記憶装置11に記憶されたチャネル・プログラムを実行する。こうしたチャネル・プログラムは、それぞれIBM製造のいわゆる370アーキテクチャのコンピュータで実施されている、チャネル制御ワード(CCW)のリストである。こうした通常のチャネル・プログラムの実行は、チャネル・プロセッサ12のチャネル実行部13で表わされる。チャネル・プロセッサ12は、FBA装置またはDASD16に接続されたFBAアダプタ15に通常のケーブル配線を介して接続されている。FBA様式によるDASD16との間でのデータの記憶及び検索は、周知の周辺データ記憶技術による。DASD16は、複数の同時回転ディスク17を含み、それらのディスクはそれぞれ、周知のように、1つまたは2つの記録面をもち、そこに多数のアドレス可能レコード・トラックがシリンダとして配列されている。

上位プロセッサ10によって実行されるプログラムの多くは、周知のCKDデータ様式を使用す

- 13 -

様式のブロックが第2の様式のアドレス可能部分となる仮想トラックとしてFBA様式でエミュレートされる。この実施例は、データの多くの仮想トラックで構成される仮想CKDディスクをエミュレートすることを含む。

本発明の上記及びその他の目的、特徴及び利点は、添付図面に示す、本発明の好ましい実施例についての以下の具体的な説明から明らかになるであろう。

#### D. 実施例

図面を参照すると、同じ部品及び構造上の特徴を同じ番号と文字で示す。第1図に、本発明を利用すると好都合な情報処理システムが、主記憶装置11からのプログラムを実行する上位プロセッサ10を含むものとして示されている。周辺プログラム記憶装置(図示せず)からのプログラムを主記憶装置11にページングするには、通常の手順を使用する。(通常は上位プロセッサ10と同じキャビネットに実装されている)チャネル・プロセッサ12が、上位プロセッサ10の望ましい

- 12 -

るので、こうしたCKD様式データはDASD16に直接記録できず、通常CKD様式のDASDが使用される。上位プロセッサ10のプログラムをFBA様式のDASD16に関連して良好に動作させるため、チャネル実行部13とFBAアダプタ15の間にエミュレータ20が挿入され、CKD様式化データをFBA様式でエミュレートする。エミュレータ20は半導体チップの形の論理回路でよく、ROM形式、ディスク形式、または周辺プログラム記憶装置からチャネル・プロセッサ12へのページングによってプログラミングされる。こうしたエミュレータは、チャネル・プロセッサ12の回路によって実行されるように主記憶装置11に格納してもよく、チャネル・プロセッサ12の個別プログラム・メモリに格納してもよい。

CKDデータは仮想データ・トラックとしてのFBA様式のDASD16に記憶される。CKDデータは1つまたは複数のCKDデータ・トラックにある。仮想トラック中のバイト数は、CKD

- 14 -

装置またはFBA装置の物理トラック・サイズには依存せず、CKD様式データのサイズによって仮想トラックのサイズが決まる。CKDデータ中のバイト数は既知、または容易に計算できる(上記ボール論文参照)。以下の説明では、DASD 16のディスク17がCKD様式データを受け取るようにフォーマットされていると仮定する。前記の「CKDエミュレーションのためにFBAディスクを初期化するステップ」の流れ図を参照。エミュレーションを作成する最初のステップは、データを記憶するのに必要なDASD 16のトラックのブロック数を計算することである。CKDデータのバイト数をXとし、DASD 16の1つのアドレス可能ブロックに記憶できるバイト数をBPBとする。さらに、エミュレートされたトラックまたは仮想トラックのCKDレコードへの直接アクセスを容易にするため、各ブロックに制御情報ヘッダHDを記録することが望ましい。ヘッダ用に使用されるバイト数をCIDとする。次に、FBA様式のB個のブロック(Bは $(X / (BPB -$

- 15 -

仮想トラック・バッファ26は、主記憶装置11の一部として割り振られる。新しい大型の上位プロセッサでは、チャンネル・プロセッサ12はそれ自体のメモリをもつ。その場合は、バッファ26はチャンネル・プロセッサ12のメモリの一部として割り振られる。各項目領域27は、FBAブロックの記憶容量に等しい記憶容量をもつ。各項目の一番左の2つのバイトは、後で説明する制御情報ヘッダHDを記憶し、各バッファ項目の残りのバイトはFBA様式にしたCKDデータを記憶する。HDは、各FBAブロックのバイト位置0と1に記録される。CKDデータは、各FBAブロックのバイト位置2以降に記録される。エミュレータ・プログラム25は、第3図に示すバッファ26の項目に当てはまるようにCKDデータを処理する。エミュレーションは、オリエンテーション・テーブル28を用いて仮想トラックの仮想回転オリエンテーションを行なうエミュレーション・プログラム25によって完了される。この仮想オリエンテーションは、バッファ26に記憶されたデータ

- 17 -

CID))を上整数に切り上げる。FBA様式のDASD 16のB個のブロックを、CKDデータを記憶するために割り振る。こうした割り振りは、通常の処理手順に従う。1つのFBAトラックを使用するのかそれとも複数のトラックを使用するのか、その割り振りが断片化され、断片が互いに連結してリストされるかどうか(断片化すると性能が低下する)、その割り振りがDASD 16の指標で始まるのか、それともトラック終端(EOT)で始まるかどうかは問題ではない。

CKDデータの修正は、修正の流れを示す第2図とエミュレーションのフォーマット化を示す第3図を参照すると最もよく理解できる。この説明では、エミュレータ20はチャンネル・プロセッサ12で実行できる1組のプログラム25として実施される。これらのチャンネル・プログラムを、この説明の後ろの部分で流れ図として示す。本発明は、本発明にしたがって構成されたプログラムをFBAアダプタ15など周辺サブシステム制御装置に格納することによっても容易に実施される。

- 16 -

に関するものである。CKDデータがバッファ26内でFBA様式にエミュレートされると、バッファ26の内容がFBAアダプタ15からDASD 16に送られて、割り振られたFBAブロックに記憶される。構成された実施例では、割り振られエミュレータ20でフォーマットされたFBAブロックの内容が、こうしたブロックがCKDデータを含むか否かにかかわらず、まずDASD 16からバッファ26に読み取られる。

第3図は、CKDデータが真のエミュレーションでFBAブロックに記憶されるCKD様式及びFBA様式を示す。CKDデータ様式は、トラック終端(EOT)指標30(CKD DASDでは、EOTは指標である)を含む。仮想トラックでは、EOTがトラックの始めと終わりを示す。すなわち、論理的には仮想トラックは円形である。位置31にあるレコード間ギャップG1が、EOTをCKDトラックで常に見られる第1のレコードHA 32から分離する。第2のレコード間ギャップG2、33がHA 32をR0レコード34か

- 18 -

ら分離する。R C レコード 3 4 は、制御情報 R O カウント・フィールド (R O C F) 3 5 と R O データ・フィールド R O D F 3 6 を含む。第 3 のレコード間ギャップ G 3 3 7 が、レコード R O 3 4 を第 1 のデータ・レコード R 1 4 0 から分離する。物理アドレス・マーク A 4 1 が相次ぐレコードの間において C K D 物理トラックのギャップ G 3 に挿入される。C K D 物理アドレス情報 A がエミュレーションから削除される。C K D レコード 1 は、カウント・フィールド C F 4 2、キー・フィールド K F 4 3 及びデータ・フィールド D F 4 4 を含む。K F 4 3 は任意選択のフィールドである。レコード R 1 4 0 中の制御情報は、C F 4 2 と K F 4 3 に含まれている。C K D データ中の後続のすべてのレコード R 2 などは図示した周知の様式を使用する。

エミュレーションのための C K D データを記憶する F B A 様式トラックが、F B A ブロック 5 0 及びそれに続く F B A ブロック 5 1 として示されている。ブロック 5 0 は、C K D データの第 1 の

- 19 -

ングは C K D トラック上のレコードの回転位置であることに留意されたい。仮想トラックは F B A で物理的に実現されているので、真のエミュレーションには、この同じバイト変位が仮想トラックで発生することが必要である。ブロック 5 0 は、その最初の 2 つのバイト 0 と 1 が制御ヘッダ H D 5 5 で占められている。各 H D は、当 F B A ブロックに記憶される最初に発生したカウント・フィールドの第 1 バイトを指す、バイト変位ポインタ 5 6 を含む。F B A ブロック 5 0 中で、ポインタ 5 6 は R O C F、すなわちレコード R O 3 4 のカウント・フィールド 3 5 の第 1 バイトを指す。ある F B A ブロックが C K D カウント・フィールドを記憶していない場合、ポインタはヌルで、その F B A ブロックでレコードが始まらない (カウント・フィールドなし) ことを示す標識ビット 5 8 が設定される。この説明の範囲外の他の制御情報も H D 5 5 に含まれる。2 バイト (F B A ブロックの始めから数えて 3 バイト) 目からは、ギャップ G 1 3 1 が始まる。このレコード間ギャップは、

- 21 -

部分を記憶するものとして示されている。ブロック 5 0 は、F B A トラックのどの円周上の位置にあってもよい。F B I D (固定ブロック物理識別) 5 2 が F B A ブロック 5 0 と 5 1 の間にある。F B I D 5 2 は後続のブロックのトラック・アドレスとブロック・アドレスを記憶する。F B I D 5 3 は F B A ディスク上のブロック 5 0 の位置を物理的に識別する。この F B A 物理識別は、D A S D 上で C K D 様式で使用される C K D 物理識別と置き換わる。こうした置換えにより、物理 D A S D 位置を利用する回復処理が可能になる。後で明らかになるように、データの始めからのバイト変位で表わした C K D レコード・アドレスは、F B A 物理アドレスと所定の既知の関係をもち、F B A 様式ディスクに記憶された C K D レコードへの直接かつランダムなアドレッシングが可能になる。

F B A ブロック 5 0 は、C K D レコードの C K D バイト変位アドレッシングを維持するための C K D 様式データのバッキングの例である。C K D 様式ディスク上での C K D バイト変位アドレッシ

- 20 -

レコード H A 3 2 の最初のバイトが F B A ブロック 5 0 内で、H A 3 2 が C K D 様式で E O T 3 0 からずれているのと同じバイト変位で始まることを示す、正しい数のギャップ・バイト (例えばゼロ) をもつ。C K D レコード間ギャップ 3 3 は、F B A ブロック 5 0 でも H A 3 2 と R O レコード 3 4 の間で同様に複製されており、R O C F 3 5 の第 1 バイトが、F B A ブロック 5 0 のバイト 2 (仮想 C K D トラックの始め) から、C K D 様式で E O T からずれているのと同じ変位をもつ。後続のすべてのレコード間ギャップ 3 7 と 6 3 も同様に、C K D トラックの始め及び仮想トラックの始めからの各 C K D カウント・フィールドの第 1 バイトのバイト変位を実現するようなサイズになっている。実際の数値計算は上記のボール論文に記述された 2 つの様式から容易に明らかになるので、簡潔にするために計算を省略する。

仮想トラックの F B A ブロックに対する物理的關係は比較的単純である。各 F B A ブロックは 2 0 4 8 バイトを記憶できるものとして示されてい

- 22 -

る。HD 55に2バイトが使用されるので、各FBAブロックは仮想トラックの2046バイトを記憶する。FBAブロック50はバイト0ないし2045を記憶し、次の隣接FBAブロック52はバイト2046ないし4091を記憶し、第3のFBAブロック(図示せず)はバイト4092ないし6137を記憶し、以下同様にして、すべての後続FBAブロック(図示せず)が仮想CKDトラックとしてCKDデータを記憶する。割り振られた1組のB個の(Bは整数)FBAブロック中の最後のFBAブロックは、仮想トラックによっておそらく完全には使用されない。そうした場合、埋込みバイトが最後に割り振られたFBAブロックの未使用部分を充填する。

CKDトラック中にある制御情報の大半は削除され、物理的FBAブロックの仮想CKDトラックには記録されない。各FBAブロックは、通常それ自体の誤り検出訂正(ECC)冗長バイト、それ自体の埋込みバイト、それ自体の物理パラメータ表示(欠陥情報など)などをもつ。CKD記録

- 23 -

オリエンテーション・テーブル28は、CKDに関連するプログラムがバッファ26を用いてデータを容易にアドレスし転送できるようにする仮想回転CKDバイト変位情報を示す。テーブル28は以下に示すいくつかの項目を含んでいる。

## オリエンテーション・テーブル28

OR	方向設定ビット(論理または仮想方向設定)
CC	現シリンダ・アドレス
CH	現ヘッド・アドレス
CS	現セクタ・アドレス
CFP	現フィールド・ポインタ
CFT	現フィールド形式(カウント、キーまたはデータ)
NCP	次のカウント・フィールド・ポインタ
PCP	前のカウント・フィールド・ポインタ
DL	バイト単位のCKDデータ・フィールド長
KL	バイト単位のキー・フィールド長
NMR	レコードなしビット

- 25 -

の制御部分にある関連するDASD物理パラメータが削除され、CKDバイト・アドレスを用いてレコードへのアクセスを可能にするのに十分正確な仮想トラックを論理的に定義するのに必要なパラメータだけが保持される。CKDカウント・フィールドは28バイトから12バイトに減少される。保持される12バイトは、CCHH(シリンダ・ヘッド・アドレス)、R(レコード番号)、KL(バイト単位のキー長)及びDL(バイト単位のデータ長)を含む。さらに、仮想トラックの最後のレコードを識別するため、エミュレータ20がカウント・フィールドCFエミュレーションに1ビットを追加する。追加される1ビットLC65は、1に設定されると、カウント・フィールドが仮想CKDトラックの最後のカウント・フィールドであることを示す。仮想トラックにレコードが追加されると、1に設定されたLCビット65が最後に追加されたレコードに移る。他のすべてのLCビット65はゼロにリセットされる。LCビット65もROCF35に含まれる。

- 24 -

エミュレータ20はテーブル28を使って、実際のCKD DASD上のCKDデータへのアクセスに関連するシーク及び他のDASD物理活動を論理的にエミュレートする。第4図は、FBA DASD上で記録されエミュレートされるCKDデータにアクセスするデータ領域に関連するエミュレーション活動を示す簡略化した流れ図である。DASDでの最初の動作は、アドレスされたトラックをシークすることである。したがって、チャネル実行部13から受け取ったSEEKチャネル・コマンドに応じて、シーク・エミュレート・ステップ70(テキストによる流れ図「SEEKコマンドの実行」を参照)が実行される。エミュレータ20は、SEEKコマンドと一緒に渡されたCC及びHHパラメータを反映するようにテーブル28を更新する。チャネル実行部13は、周知のように、SEEK CCWからSEEKコマンドを生成する。第2のステップ73は、回転オリエンテーションを獲得することである(上記の場合、回転(バイト変位識別)オリエンテーションがな

- 26 -



かったのでOR=0)。矢印72で示されるSET SECTORチャネル・コマンドにより、エミュレータ20はCKD装置のSET SECTOR動作をエミュレートする(テキストによる流れ図「SET SECTORコマンドの実行」を参照)。このコマンドは、後続のデータ転送コマンドがアドレスされる回転位置(バイト変位)を識別するだけである。オリエンテーション・テーブル28のORビットは依然として0であり、この時、データはDASD16からバッファ26に移動されていない。矢印74でチャネル実行部によってSEARCH IDチャネル・コマンドが発行されると、上位プロセッサ10は、データの転送を望んでいることを示す。このシーケンスのこの時点で、エミュレータ20がDASD16にアクセスして、ステップ75で示すように、DASD16からバッファ26にデータをステージングする。SEARCH IDのID部分がCKDアドレス・パラメータの識別を与えることに留意されたい。これらのパラメータは、どのFBA

- 27 -

ブロックをステージングし続けて、バッファにおいてディスクの回転をエミュレートする。通常、上位プロセッサは第1のアドレスされたレコードから始まる多数のレコードを読み取りまたは書き込む。この動作は、DASD16を用いたFBA動作に関連する指標マークの有無に関わらず、仮想トラックの終端(CKDデータの終端)まで続く。チャネル・コマンドを受け取ると、それはデータ転送が停止するか、またはチャネル動作の終端に達したことを示す。上位プロセッサ・アクセスから独立したこの継続するステージングを矢印79で示す。

以下に、上記の動作の詳細を示すテキストによる流れ図を示す。流れ図の後の用語集で、使用した略語の意味を示す。これらの流れ図を実施するのに、どのプログラミング言語や論理ハードウェアを使ってもよい。実際の実施例では、例示したもの以外に、他のチャネル・コマンド実行コマンドを使ってマシンを完了してもよい。

- 29 -

ブロックがSEARCH IDコマンドに関連するCKDカウント・フィールドCFを記憶するかを識別するもの以外は、独立したアドレスには変換されない。この計算は簡単なもので、数値例は示さない。

FBAブロックがバッファ26に記憶されるとすぐ、エミュレータ20はステップ78でバッファ26にアクセスして、CKD DASD上でのCKDレコードの発見をエミュレートする。所期のレコードがバッファ26中で発見されると、オリエンテーションが行なわれたことを示すDE(装置終了)が上位プロセッサ10に送られる。オリエンテーション・テーブル28のORビットが1に設定される。上位プロセッサ10は、矢印77で示されるデータ転送コマンドを発行し、それにより、エミュレータ20がステップ78でバッファ26と主記憶装置11の間でデータを転送することによりDASDから読み取られたCKDをエミュレートする。こうしたデータ転送中に、エミュレータ20はDASD16からバッファ26にFBA

- 28 -

#### エミュレータ2.0のプログラム25を示す流れ図

以下に示すテキストによる流れ図は、本発明を例示するチャネル・プロセッサ12のマシン・ステップを示す。これらのステップは、完全なマシン設計を記述するものではない。完全なマシン中で見られる他のプログラムや構造は、本発明を実施するのに必要ではないからである。

#### 呼び出されるサブルーチンを含むチャネル・コマンド実行の流れ図

##### SEEKコマンドの実行

受信コマンドの妥当性検査を行なう。

ファイル・マスクでシークが可能なことを確認する。

受信コマンドからシーク・パラメータを読み取り、様式を確認する。

シリンドラ(CC)アドレスとヘッド(HH)アドレスの妥当性検査を行なう。

シークが必要な(現CCHHが要求されたCCHHに一致しない)場合は続行し、そうでない場合

- 30 -

は「チャンネルにCEDE」ステップに進む。  
CCHHをセーブし、それをシーク回路に入力する。

「シーク必要」ビット100をNS=1に設定する。

「オリエンテーション・ビット」をOR=0に設定する。

チャンネルにCEDEする；連鎖を検査して、次に進む。

#### SET FILE MASKコマンドの実行

受信コマンドの妥当性検査を行なう。

受信コマンドからファイル・マスク・パラメータを読み取る。

ファイル・マスクをセーブする。

「オリエンテーション・ビット」をOR=0に設定する。

チャンネルにCEDEする；連鎖を検査して、次に進む。

#### SET SECTORコマンドの実行

受信コマンドの妥当性検査を行なう。

- 31 -

ションが行なわれ)かつ多トラック動作でない場合、ケースIVを実行する。

(最後のレコードを越えて、または最後のレコードの非カウント・フィールドにまでオリエンテーションが行なわれ)かつ多トラック動作が実行されている場合、ケースVを実行する。

5つのケースの終わり

SIZE=受信パラメータのバイト数に設定する。  
PTRから始めてSIZEをパラメータと比較する。

比較の成否の結果をチャンネルに送る；

比較が成功で、SIZE<5の場合、SIZE  
5としてセーブする。

状況の終了を行なう。

ケースIのステップ

PTR=CFP

CFP=CFP+12

KL=0の場合はCFT=DF、そうでない場合はCFT=KF

受信コマンドからパラメータを読み取る。

コマンドのセクタ番号の妥当性検査を行ない、それをSとしてセーブする。

「オリエンテーション・ビット」をOR=0に設定する。

チャンネルにCEDEする；連鎖を検査し、次に進む。

#### SEARCH IDコマンドの実行

受信コマンドの妥当性検査を行なう。

OR=0の場合、「GET ORIENTED」サブルーチン呼び出す。

受信コマンドからパラメータを読み取る。

次の5つのケースの1つを実行する。

CFT=CONTの場合、ケースIを実行する。

CFT=ROCONTの場合、ケースIIを実行する。

IDが最後のレコード以外のレコードの非カウント・フィールドを指す場合、またはHAとROを指すIDが存在する場合、ケースIIIを実行する。

(最後のレコードを越えて、または最後のレコードの非カウント・フィールドにまでオリエンテ

- 32 -

ケースIIのステップ

PTR=CFP

CFP=CFP+12

KL=0の場合はCFT=RODF、そうでない場合はCFT=ROKF

ケースIIIのステップ

PTR=NCP

KL=ポインタPTRでのレコードのキー長

DL=ポインタPTRでのレコードのデータ長

CFT=HAフィールドの場合はPCP=CFP、そうでない場合はPCP=次のカウント(次のカウント・ポインタを計算する)

CFP=PTR+12

KL=0の場合はCFT=DF、そうでない場合はCFT=KF

NCP=次のカウント・ポインタを計算する(サブルーチン呼び出す)

LC=1の場合はNMR=真(1)、そうでない場合はNMR=偽(0)

- 33 -

- 34 -

## ケースⅣのステップ

EOT=1の場合はERROR、そうでない場合はEOT=1(指標通過)に設定する

$BB = (T * C + H) * B$

$BBLAST = (T * C + H) * B + (BPS * S / BBB) - 1$  (用語集のBBBの項を参照)

Sを $> G1 / BPS$ かつ $< (G1 + G2' + SIZE(HA) + PADH + ECCH) / BPS$ に設定する。

ステップ4でGET ORIENTEDを呼び出す。

GET ORIENTED=0の場合、レコードが見つからないことを示し、そうでない場合はレコードが見つかったことを示す。

$PTR = CFP$

$CFP = CFP + 12$

KL=0の場合はCFT=RODF、そうでない場合はCFT=ROKF

## ケースⅤのステップ

H = H + 1に設定する。新しいH > Tの場合、誤

- 35 -

ションが行なわれ、PO=1の場合、ケースAを実行する。そうでない場合は、

ROに対してオリエンテーションが行なわれ、かつ(前のCMDが探索IDでも探索キーでもなく、R1が存在する)場合、ケースBを実行する。そうでない場合は、

HAに対してオリエンテーションが行なわれ、かつROとR1が存在する場合、ケースCを実行する。そうでない場合は、

EOT=1で、多トラック動作でない場合、ケースDを実行する。そうでない場合は、

EOT=1で、多トラック動作の場合、ケースEを実行する。

## ケースの終わり

位置PTRからバイト数Lを読み取る。

L=0の場合、CEDEでチャンネルにUCし、そうでない場合はチャンネルにCEDEする。

## ケースAのステップ

$PTR = CFP$

CFT=KFまたはROKFの場合、PTR=P

- 37 -

りを知らせる。

$S > (G1 / BPS)$  かつ  $< (G1 + G2' + SIZE(HA) + PADH + ECCH) / BPS$  に設定する。

ステップ2から始めて「GET ORIENTED」を呼び出す。

レコードが見つからない場合、戻ってSを設定する。そうでない場合は続行する。

$PTR = CFP$

$CFP = CFP + 12$

KL=0の場合はCFT=RODF、そうでない場合はCFT=ROKF

## 読取りデータ・コマンドの実行

受信コマンドの妥当性検査を行なう。

前のCMDが「SEARCH」で、比較して等しかった場合、PO=1

OR=0の場合、GET ORIENTEDを呼び出す。

非ROレコードに対してオリエンテーションが行なわれている(またはROに対してオリエンテ

- 36 -

TR+KL。そうでない場合は、

$CFT = CNT$  または  $ROCF$  の場合、 $PTR = PTR + 12 + KL$

ステップYを実行する。

## ステップY

$L = DL$

$CFP = NCP$

PCP=PCPから次のCNTを計算する。

NMR=1の場合、CFT=指標(EOT)、そうでない場合はループを実行する。

## ループ実行

$CFT = CNT$

$KL = CFP$  の KL

$DL = CFP$  の DL

$NCP = CFP$  の CNT から計算する

LC=1の場合はNMR=1、そうでない場合はNMR=0

## ケースBのステップ

$CFP = NCP$

$L = CFP$  の DL

- 38 -

$PTR = CFP + 12 + KL$

$PCP = CFP$

$CFP = CFP$  の  $CNT$  から次の  $CNT$  を計算する

ステップ X を実行する

ステップ X

$PCP$  について  $LC = 1$  の場合、 $CFT = EOT$ 、  
そうでない場合はループを実行する。

ループ実行

$CFT = CNT$

$KL = CFP$  の  $KL$

$DL = CFP$  の  $DL$

$NCP = CFP$  から次の  $CNT$  を計算する

$LC = 1$  の場合は  $NMR = 1$ 、そうでない場合は  $NMR = 0$

実行を終了する。

ケース C のステップ

$PTR = NCP$  から次の  $CNT$  を計算する

$L = PTR$  の  $DL$

$CFP = PTR$

- 39 -

ステップ Y を実行する。

ケース E のステップ

$H = H + 1$  と設定する。新しい  $H > T$  の場合、誤りを設定する。

$S > (G1 + G2' + SIZE(HA) + PADH + ECCH) / BPS$  に設定する。

$S < (G1 + G2' + SIZE(HA) + PADH + ECCH + G2 + G3 + SIZE(CF) + PADC + ECCC + 8 + PADD + ECCD) / BPS$  に設定する。

$GET ORIENTED$  を呼び出す。ステップ 2 から始める。

$NRF = 1$  の場合 (レコードが見つからなかった場合)、ケース E の第 1 ステップに戻る。そうでない場合は続行する。

$PTR = CFP + KL + 12$

ステップ Y を実行する。

ケース A ないし E の終わり

$PTR$  から始めて  $L$  バイトをチャネルに送る。

$L = 0$  の場合、 $UC$  をチャネルに  $CEDE$  する。

- 41 -

$PTR = PTR + 12 + KL$

$PCP = CFP$

ステップ X を実行する。

ケース D のステップ

$EOT = 1$  の場合、 $ERROR$  を設定する。そうでない場合は  $EOT = 1$  に設定する

$BB = (T * C + H) * B$

$BBLAST = (T * C + H) * B + (BPS * S / BBB) - 1$  (BBB - 用語集を参照)

$S > (G1 + G2' + SIZE(HA) + PADH + ECCH) / BPS$  に設定する

$S < (G1 + G2' + SIZE(HA) + PADH + ECCH + G2 + G3 + SIZE(CF) + PADC + ECCC + 8 + ECCD) / BPS$  に設定する。

$GET ORIENTED$  を呼び出す。ステップ 4 から始める。

$OR = 0$  の場合、レコードが見つからなかった。そうでない場合はレコードが見つかった。

$PTR = CFP + KL + 12$

- 40 -

そうでない場合はチャネルに  $CEDE$  する。

WRITE DATA コマンドの実行

受信コマンドの妥当性検査を行なう。

$CKD$  データを上位プロセッサから受け取り、主記憶装置 11 の個別バッファ領域に記憶する。以下のステップは、記憶された  $CKD$  データが第 3 図に示すように変換された後でそのデータに対して行なわれる。

$PTR = CFP$

$CFT = KF$  または  $ROKF$  の場合、 $PTR = PTR + KL$

ステップ Y を実行する。

$PTR$  で上位プロセッサからの  $CKD$  データの  $L$  バイトをバッファ 26 に記憶する。

更新されたまたは新しい  $FBA$  ブロック (50 や 51 など) を  $DASD 16$  に書き込む。

すべての  $FBA$  ブロックを  $DASD 16$  に書き込むと、チャネルに  $CEDE$  を送る。別の動作に進む。

- 42 -

GET ORIENTED サブルーチン

このサブルーチンはオリエンテーション・テーブル 28 を管理する。

このコマンド連鎖中に SEEK コマンドがない場合、 $CC = CH = 0$

このコマンド連鎖中に SET SECTOR コマンドがない場合、 $CS = 0$

## ステップ 2

バッファ 26 にステージングされる、すなわち転送される最初のレコードを記憶する最初の FBA ブロックを識別する。

最初の FBA ブロックのアドレス =  $(T * C + H) * B + (BPS * CS / BBB)$  下の整数に切り下げる。

DASD 16 からバッファ 26 にステージングされる最後の FBA ブロックを識別する。

最後の FBA ブロックのアドレス =  $(T * C + B) + (B - 1)$

## ステップ 4

上記で識別された FBA ブロック及びすべての中

- 43 -

$BPS * S \leq Z$  の場合、HA または カウント・フィールドに対して方向設定される（セクタ S を越えた場合は、ステップ 9 に飛ぶ。）そうでない場合は次に進む。

$LC = 0$  の場合、

$Z' = Z + 12 + KL + DL + PAD C + ECC C + PAD D + ECC D + G2 + G3$  に設定する。

KL がゼロでない場合は  $Z = Z' + PAD K + ECC K + G2$ 、そうでない場合は  $Z = Z'$

（上記では、最初のレコードと EOT の間で探索が行なわれると仮定する）

## ステップ 9

$NS = 0$ （シークする必要はない）と設定する。方向設定テーブル 28 を更新する。

$CFP = Z$

KL = レコードのカウント・フィールドの KL

DL = レコードのカウント・フィールドの DL

CC = レコードのカウント・フィールドのシリンダ・アドレス

- 45 -

間の FBA ブロックを、第 2 図に示すようにバッファ 26 に読み込む。HD を FBA ブロック中のデータ（バイト 2-2048）から分離しておく。

最初の FBA ブロックの HD を検査する；

FBA ブロック中に CF を指すポインタがない場合、最初のカウント・フィールドが見つかるまで後続の FBA ブロック HD を検査する。

カウント・フィールドが見つからずに EOT に達した場合、仮想トラックの始めから最初の FBA ブロックまで FBA ブロックをステージングして、カウント・フィールドを見つける。カウント・フィールドが見つからない場合、チャンネルに UC を合図する。

FBA ブロック NR（最初のカウント・フィールドをもつブロック）= 最初の FBA ブロッカー  $(T * C + H) * B$  と設定する。

$Z = (NR * BBB) + HD$  からのオフセット・ポインタ（最初のカウント・フィールドの第 1 バイトが FBA ブロック NR 内にあるバイト数）と設定する。

- 44 -

CH = レコードのカウント・フィールドのヘッド・アドレス

$CF T = S * BPS < G1$  ならば HA、そうでない場合は、 $G1 < S * BPS < G1 + G2' + SIZE(HA) + PAD H + ECC H$  ならば R0 カウント・フィールド R0 CF、そうでない場合はデータ・レコード「n」の CF。

OR = 1

$LC = 1$  の場合、 $NMR = 1$  と設定する。そうでない場合は  $NMR = 0$

呼出しプログラムに戻る。

次のカウント・フィールド・ポインタの計算のサブルーチン

ポインタ PTR 2 はエミュレートされた仮想トラック中の次のカウント・フィールド位置を示す。

$PTR = G1$  の場合、 $PTR 2 = PTR + 12 + PAD H + ECC H + G2'$ 、そうでない場合は  $PTR 2 = PTR + 12 + (PTR \text{ が指すレコードの }) KL + (PTR \text{ が指すレコードの }) DL + PAD C + ECC C + PAD D + ECC$

- 46 -

$D + G2 + G3$

(PTRが指すレコードの)  $KL \neq 0$  の場合、  
 $PTR2 = PTR2 + G2 + PADK + ECC$   
 $K$

PTR2を呼出しプログラムに戻す。

エミュレーションのためにFBAディスクを初期化するプログラム

CKDディスクのFBAブロックの数  
 $X = CKD$ トラックのバイト数と置く。

$B = X / BBB$  (各CKDトラック中で使用されるブロックの数) と置く。

CKD装置1台当たりのトラック数は、シリンダの数とシリンダ1個当たりのトラック数の積である。各CKD装置のブロックの数は、CKD装置のトラック数とBの積である。

CKDエミュレーションのためにFBAディスクを初期化するステップ

$CYL =$ エミュレートされるCKD装置のシリンダ数と置く。

まずエミュレーションに使用されるFBAディス

- 47 -

= 8及びLCがすべて1である点以外は上記のHAの場合と同様にR0を作成する。

FBAディスクのFBAディスク上でB-1個のブロックで分離された適切なブロックに現バッファからのブロックを書き込む。

CKDトラックをB個のFBAブロックに書き込むステップ

この流れ図は、CKDトラックからFBAディスク上のB個のFBAブロックにデータをコピーすることに関連している。

バッファ項目(番号0ないしB-1)でバッファ26を構成する。

初期化された主記憶装置のBフラグ・ビットBF101を0(偽)に設定する。所定のFBAブロック(バッファ26項目)について、そのフラグBF=1の場合、そのフラグ・ビットに関連するFBAブロックに制御情報が記録されている。バッファ26中で、当該のすべてのHDの第2バイトに2進パターン00001000を記憶する。このパターンは、データが当該のFBAブロックに

- 49 -

クのすべてのFBAブロックを初期化して、使用される各FBAブロックのバイト0と1に16進数0008を記憶させる。さらに、B番目ごとのブロックが論理HAとR0レコードを含む。この含有について以下に記述する。

HAとR0で初期化されるFBAブロックのすべてについて実行する。

バッファ26を、繰り返して使用されるFBAブロックに設定する。

バイト0と1に、最初の12ビットのG1と最後の4ビットのすべての0を記憶する。

各ブロックのG1+2のバイト・オフセットの所に、HAを次のように書き込む；フラグ・バイトを0、CC=0からCYL-1まで増分されるシリンダ番号、HH=ゼロからT-1まで増分されるトラック番号、Rバイト(レコード番号)=0、KL=0、DL=0、LC=0、パッド・バイト=0

$G1 + G1' + 2(HA) + ECC H + PAD H$ の各ブロックのバイト・オフセットの所に、DL

- 48 -

記録されていないことを示す。

CB102は、バッファ26項目に書き込まれる次のバイトを指す。最小値は2である(HBバイト0と1を飛び越す)。

CKDトラックは、フィールドHA、ROCF、RODF、CF1、KF1、DF1などをもつ主記憶装置中のCKDトラックとしてコピーされている。

メモリ・ポインタはすべてゼロである。

カウント・フィールドは28バイトではなく12バイトしかない。CKD物理パラメータなどが削除されている。

書き込みが呼び出される度に、CBから始めてバイトが書き込まれ、HDバイトは常に飛び越される。カウント・フィールドが書き込まれ、それに対応するフラグBFがオフの場合、そのフラグBFはオンになる。対応するHDは、書き込まれたカウント・フィールドに対するオフセットを含むようにされる。CBは常に更新される。

- 50 -

CKDをバッファ項目に書き込むステップ

CB = 2

G1 ゼロ・バイトを書き込む（各バイトが記録されるときCBを更新する）。

HAを書き込む。

G2' + SIZE (HA) - 12 + PADH + ゼロ・バイトのECC Hを書き込む。

ROCFを書き込む。

RODFを書き込む。

G2 + G3 + SIZE (CNT) - 12 + PAD C + ECCC + PADD + ゼロ・バイトのECC Dを書き込む。

レコードR0以外の各CKDレコードについて繰り返す。

CFを書き込む。

KFを書き込む。

DFを書き込む。

G1 + G2 + G3 + SIZE (CNT) - 12 + PAD C + ECCC + PAD K + ECCK + PADD + ゼロ・バイトのECC Dを書き込む。

- 51 -

CE チャンネル終了信号、チャンネルが開放されることを示す

CEDE チャンネル終了、装置終了、チャンネル動作の完了を示す

CF CKD様式レコードのカウント・フィールド

CFP 現フィールド・ポインタ

CFT 現フィールド形式

CID FBA装置中のヘッダのバイトの数

CKD カウント、キー、データ

CMD チャンネル・コマンド

CYL 1つのCKD DASD中のシリンダ数

DE 装置終了、装置が動作を完了したことをチャンネルに合図する

DF CKD様式レコードのデータ・フィールド

DL データ・フィールドDFの長さ、カウント・フィールドCF中のフィールド

ECC エラー検出・訂正冗長バイト

ECCC カウント・フィールドCF中のECC バイトの数

- 53 -

DASD16のB個のFBAブロックにバッファ28を書き込む。

用語集

B 1つのCKD仮想トラックに含まれるFBAブロックの数

BB 現バッファにステージングされる最初のFBAブロックのブロック番号。BBB: 値BPB-CID

BBLAST CKDステージング動作で現バッファにステージングされる最後のブロックのFBAブロック番号

BF フォーマットされたかどうかを示すブロックbに関連するフラグ

BPB FBAブロック1個当たりのバイト数

BPS CKDデータ様式のセクタ1個当たりのバイトの数

CB 処理中の現バイトを指すポインタ

CC エミュレートされる現バイトのシリンダ番号またはアドレス

CCW チャンネル・コマンド（制御）ワード

- 52 -

ECCD データ・フィールドDF中のECC バイトの数

ECCH HA中のECCバイトの数

ECCK キー・フィールドKF中のECCバイトの数

EOT トラック終端、物理DASDの指標に関連する

FBA 固定ブロック・アーキテクチャ

Gx バイト中のギャップ長、ただしx = 1ないし3、CKD様式

HH DASD (CKDまたはFBA) のヘッドまたは表面アドレス

HA CKD様式のホーム・アドレス・レコード

HD CKDエミュレーション用のFBAブロックのヘッダ（制御フィールド）

KF CKD様式のキー・フィールド

KL キー長、KLを示すカウント・フィールド中CFの標識

NCP 次のカウント・フィールド・ポインタ

NR アクセスされる次のCKDレコード

- 54 -

N R F 探索基準に合致するレコードが見つからない

N S ビットまたはフラグをシークする必要がある

P A D C カウント・フィールド C F 中の埋込みバイトの数

P A D D データ・フィールド D F 中の埋込みバイトの数

P A D H H A 中の埋込みバイトの数

P A D K キー・フィールド K F 中の埋込みバイトの数

P C P 前のカウント・フィールド・ポインタ

P D S 区分データ

P O 以前に論理的にオリエンテーションが行なわれた

P T R 処理中のフィールド／項目を指すのに使われるポインタ

R レコード番号、フィールドはカウント・フィールド C F 中にある

S セクタ番号

- 55 -

1 0 ……上位プロセッサ、1 1 ……主記憶装置、  
1 2 ……チャネル・プロセッサ、1 3 ……チャネル  
実行部、1 5 ……F B A アダプタ、1 6 ……D  
A S D、2 0 ……エミュレータ、2 5 ……エミュ  
レータ・プログラム、2 8 ……仮想トラック・バッ  
ファ、2 8 ……オリエンテーション・テーブル。

出願人 インターナショナル・ビジネス・  
マシーンス・コーポレーション

代理人 弁理士 山 本 仁 朗  
(外 1 名)

- 57 -

S I Z E エミュレートされる C K D 装置のバイトで表わしたフィールド（次の項目で識別されるフィールド）のサイズ

T D A S D の 1 つのシリンダのトラック数

U C 単位検査、誤りを示すチャネルへの信号

V T O C D A S D の内容のボリューム・テーブル

#### E. 発明の効果

本発明は、エミュレートされるレコードのアクセス性能を改善する。

#### 4. 図面の簡単な説明

第 1 図は、本発明が有利に実施できる情報処理システムの簡略化した図である。

第 2 図は、第 1 図に示した情報処理システムで本発明を実施する構成図である。

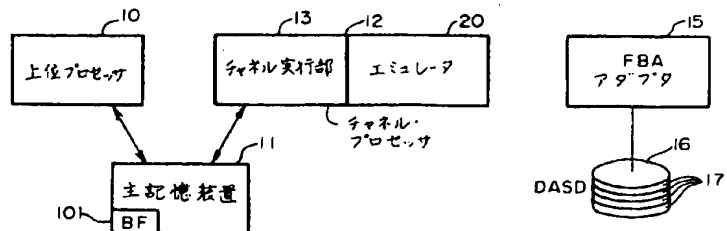
第 3 図は、第 1 と第 2 のデータ様式及び第 2 図に示すように実施された第 2 のデータ様式で第 1 のデータ様式のエミュレーションの構成図である。

第 4 図は、第 2 図で使用するマシン動作の簡略化した流れ図である。

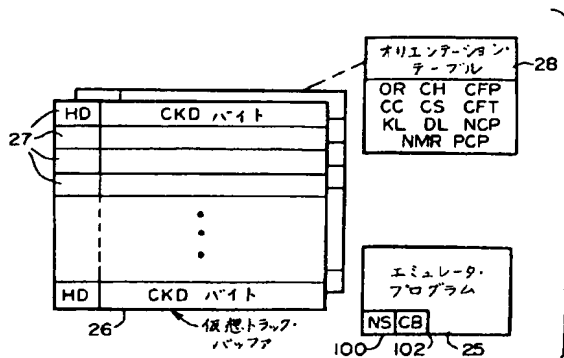
- 56 -



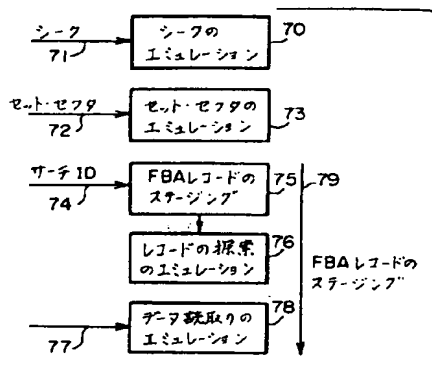
第 1 圖



第 2 図



第 4 図



第 3 図

